CS335a: Computer Networks

Professor: Maria Papadopouli
TA: Giorgos Mellios (csdp1395@csd.uoc.gr)
Deadline: 03/01/2026

SUBMISSION GUIDELINES

- **Your report should be in PDF format.**
- **Please submit your assignment via the e-learn platform.**
- **The maximum grade you can get is 120 with 20 out of the 120 points being BONUS.**
- **To prevent plagiarism, in each assignment series a random sample of students will be selected for further oral examination.**

Assignment 5: MAC Layer

Exercise 1 (20pts)

i) (5p) Explain the difference between **Error Detection** and **Error Correction**. Why might a link-layer protocol choose to only detect errors (and discard the frame) rather than correcting them? Provide an example scenario for each approach (detection vs. correction).

ii) (5p) Consider the following grid of data bits with **Even Parity**:

				Row Parity	
0	1	1	0	**0**	
1	1	0	1	**1**	
0	0	0	0	**0**	
1	0	1	1	**1**	
Col Parity	**0**	**0**	**0**	**0**	**0**

1. Suppose the bit at Row 2, Column 3, originally "0", is flipped to "1" during transmission.
2. Redraw the grid as received (with the error).
3. Explain exactly how the receiver detects the error and identifies which specific bit needs to be corrected using the row and column parity bits.

iii) (10p) Consider the data bit sequence D = 1 0 1 0 1 0 1 0 and the generator polynomial G = 1 0 0 1 (3 bits + 1).

1. Calculate the **CRC bits (R)** that the sender should append to the data.
2. Show your polynomial division steps clearly.
3. What is the final transmitted bit sequence?

Indicative Answers:

i.

Difference between Error Detection and Error Correction

Difference: Error Detection allows the receiver to determine if a frame has been corrupted (contains bit errors) but not necessarily which bits are wrong. Error Correction (often called Forward Error Correction or FEC) allows the receiver to detect errors and determine exactly which bits were flipped, enabling it to fix the frame without retransmission.
Correction requires more redundant bits (overhead) than simple detection. In highly reliable channels (like fiber optics), bit errors are rare. It is more efficient to use a lightweight detection code (like CRC) and simply ask for a retransmission (ARQ) on the rare occasion an error occurs, rather than burdening every single frame with heavy error-correction data.

Examples:

Detection: Ethernet or Wi-Fi (802.11) uses CRC to detect errors. If a check fails, the frame is discarded and often retransmitted.

Correction: Satellite communications or Deep Space probes. The latency (propagation delay) is so high that retransmissions are too slow, so the receiver must be able to fix errors on the spot.

ii. 2D Even Parity

1. Redrawn Grid (with error): The bit at Row 2, Column 3 flipped from 0 to 1.

				Row Parity	
0	1	1	0	**0**	
1	1	1	1	**1**	
0	0	0	0	**0**	
1	0	1	1	**1**	
Col Parity	**0**	**0**	**0**	**0**	**0**

(Note: The original parity bits remain transmitted as they were calculated before the error. The Receiver calculates new sums based on received bits.)

2. Detection and Identification:

- **Row Check:** The receiver sums the 1s in Row 2: 1+1+1+1 = 4. However, the parity bit for that row is 1. This results in an odd total (5 ones), violating the Even Parity rule. The receiver flags Row 2 as containing an error.
- **Column Check:** The receiver sums the 1s in Column 3: 1+1+0+1 = 3. The parity bit is 0. This results in an odd total (3 ones), violating Even Parity. The receiver flags Column 3 as containing an error.
- **Correction**: The intersection of the failing Row 2 and the failing Column 3 pinpoints the specific bit that flipped. The receiver flips it back to restore the data.

iii.

10101010 divided by 1001

Binary form (added zeros): 10101010000 divided by 1001

```
-----------
10101010000
1001
----
 0111010000
 0000
 ----
```

```
 111010000
 1001
----
 11110000
 1001
 ----
  1100000
  1001
  ----
   101000
   1001
   ----
    01100
    0000
    ----
     1100
     1001
     ----
      101
```

Transmitted value is: 10101010101
Remainder is 101

Exercise 2 (25pts)

i) (5p) Compare channel partitioning protocols (**TDMA**/**FDMA**) with random access protocols (**ALOHA**, **CSMA**) in terms of:

1. Efficiency at low load.
2. Efficiency at high load.
3. Fairness.

ii) (10p) Suppose that nodes A and B are on the same 10 Mbps broadcast channel, and the propagation delay between the two nodes is 245 bit times. There is no other device on that channel. Nodes A and B send Ethernet frames at the same time. As we know the frames will

collide. Assume that A and B choose different values of K in the **CSMA/CD** algorithm. Assuming no other nodes are active, can the retransmissions from A and B collide? For our purposes, it suffices to work out the following example. Suppose A and B begin transmission at t = 0 bit times. They both detect collisions at t = 245 bit times. Suppose KA = 0 and KB = 1. At what time does B schedule its retransmission? At what time does A begin transmission? **(Note: The nodes must wait for an idle channel after returning to Step 2—see protocol.**) At what time does A's signal reach B? Does B refrain from transmitting at its scheduled time?

iii) (10p) Consider N devices that use the (pure) **ALOHA** protocol to contend for a channel. The probability of retransmission probability is p. Show how to select p to optimize the utilization of the channel, considering that all N devices have a very large number of frames to transmit.

Indicative Answers:

i.

Feature	Channel Partitioning (TDMA/FDMA)	Random Access (ALOHA/CSMA)
1. Efficiency at Low Load	Low. A node must wait for its turn/slot even if no one else is sending. Bandwidth is wasted on idle slots.	High. A single node can utilize the full channel bandwidth immediately without waiting.
2. Efficiency at High Load	High. Collisions are impossible; throughput remains stable near capacity.	Low/Moderate. Collisions become frequent, leading to wasted time on retransmissions. Utilization drops (especially in ALOHA).
3. Fairness	Fair. Each node gets a dedicated, guaranteed portion of the bandwidth.	Unfair. A "lucky" node or one with aggressive timing might capture the channel. No guarantees of service.

ii.

Time, t	**Event**
0	A and B begin transmission
245	A and V detect collision
293	A and B finish transmitting jam signal
293+245=538	B's last bit arrives at A ; A detects an idle channel
538+96=634	A starts transmitting
293+512=805	B returns to Step 2 B must sense idle channel for 96 bit times before it transmits
634+245=879	A'transmission reaches B

Because A's retransmission reaches B before B's scheduled retransmission time (805+96), B refrains from transmitting while A retransmits. Thus A and B do not collide. Thus the factor 512 appearing in the exponential backoff algorithm is sufficiently large.

iii.

To calculate the probability of success, we first identify the conditions required to avoid a collision. In Pure ALOHA, devices transmit at arbitrary times. Let T be the time required to transmit one frame.

If a specific Device A starts transmitting at time t0, its frame occupies the channel from [t0, t0 + T].

A collision will occur if any other device starts transmitting:

- **Just before:** In the interval [t0 - T, t0].
- **During:** In the interval [t0, t0 + T].

Thus, for Device A to succeed, the channel must be silent for a total duration of **2T**

We focus on the success of a single device. For this device to succeed, all other **N-1** devices must remain silent during the vulnerable period (2T).

Let p be the probability a device transmits during a single frame time T.

Consequently, the probability a device is **silent** during time T is (1-p).

Since the vulnerable period is 2T (essentially two consecutive intervals of length T), a neighbor must be silent for both intervals:

$$P(\text{Neighbor Silent for } 2T) = (1-p) \cdot (1-p) = (1-p)^2$$

Since there are N-1 other devices acting independently, the probability that **all** of them are silent is:

$$P(\text{Success}) = \left[(1-p)^2\right]^{N-1} = (1-p)^{2(N-1)}$$

The channel utilization (throughput) is the total rate of traffic attempted multiplied by the probability that the traffic is successful.

- Total Attempts (Load) = $N \cdot p$

$$S(p) = \text{Load} \times P(\text{Success})$$

$$S(p) = Np \cdot (1-p)^{2(N-1)}$$

$$S(p) = Np(1-p)^{2N-2}$$

Case 1: p = 0

If p=0, no nodes ever transmit.

$$S(0) = N(0) \cdot (1-0)^{2(N-1)} = 0$$

Result: Throughput is 0.

Case 2: p = 1

If p=1, every node transmits constantly. The channel is permanently jammed with collisions.

$$S(1) = N(1) \cdot (1-1)^{2(N-1)} = N \cdot 0 = 0$$

(Note: This assumes N > 1. If N=1, then S=1, but collision logic implies N >= 2.)

Result: Throughput is 0.

Since the throughput is positive for 0 < p < 1 and zero at the boundaries, the maximum must lie in the interior interval (0, 1).

Case 3: 0<p<1

To find the maximum, we take the derivative of S(p) and find the critical point where S'(p) = 0.

$S(p) = Np(1-p)^{2N-2}$

Let $u = Np$ and $v = (1-p)^{2N-2}$.

- $u' = N$
- $v' = (2N-2)(1-p)^{2N-3} \cdot (-1)$

We apply the **Product Rule**: $\frac{d}{dp}[uv] = u'v + uv'$.

$S'(p) = N(1-p)^{2N-2} - Np(2N-2)(1-p)^{2N-3}$

$N(1-p)^{2N-2} - Np(2N-2)(1-p)^{2N-3} = 0$

Simplify

Since we are in the interval (0, 1), we know that $(1-p) \neq 0$. Therefore, we can divide the entire equation by the common term $N(1-p)^{2N-3}$.

$\frac{N(1-p)^{2N-2}}{N(1-p)^{2N-3}} - \frac{Np(2N-2)(1-p)^{2N-3}}{N(1-p)^{2N-3}} = 0$

This simplifies to:

$(1-p)^1 - p(2N-2) = 0$

$1 - p - (2Np - 2p) = 0$

$1 - p - 2Np + 2p = 0$

$1 + p - 2Np = 0$

$1 + p(1-2N) = 0$

$$p(2N-1)=1$$

$$p=\frac{1}{2N-1}$$

We have analyzed the domain [0, 1]:

1. At the boundaries (p=0, p=1), the utilization is 0.
2. At the critical point inside the interval, the utilization is positive.

Therefore, to maximize channel utilization for a finite number of devices N, the optimal transmission probability is:

$$p=\frac{1}{2N-1}$$

Exercise 3 (30pts)

i) (5p) Why do valid devices on the Internet need both a **MAC address** and an **IP address**? Describe a scenario where a device might change its IP address but keep its MAC address, or vice versa.

ii) (10p) Host A (IP: **192.168.1.10**, MAC: **AA:AA**) wants to send an IP datagram to Host B (IP: **192.168.1.20**, MAC: **BB:BB**). They are on the same **Ethernet LAN**. Host A's ARP table is empty.

1. List the steps Host A takes to obtain Host B's MAC address.
2. Specify the source and destination MAC addresses of the ARP Request frame. Is this a broadcast or unicast frame?
3. Specify the source and destination MAC addresses of the ARP Reply frame.

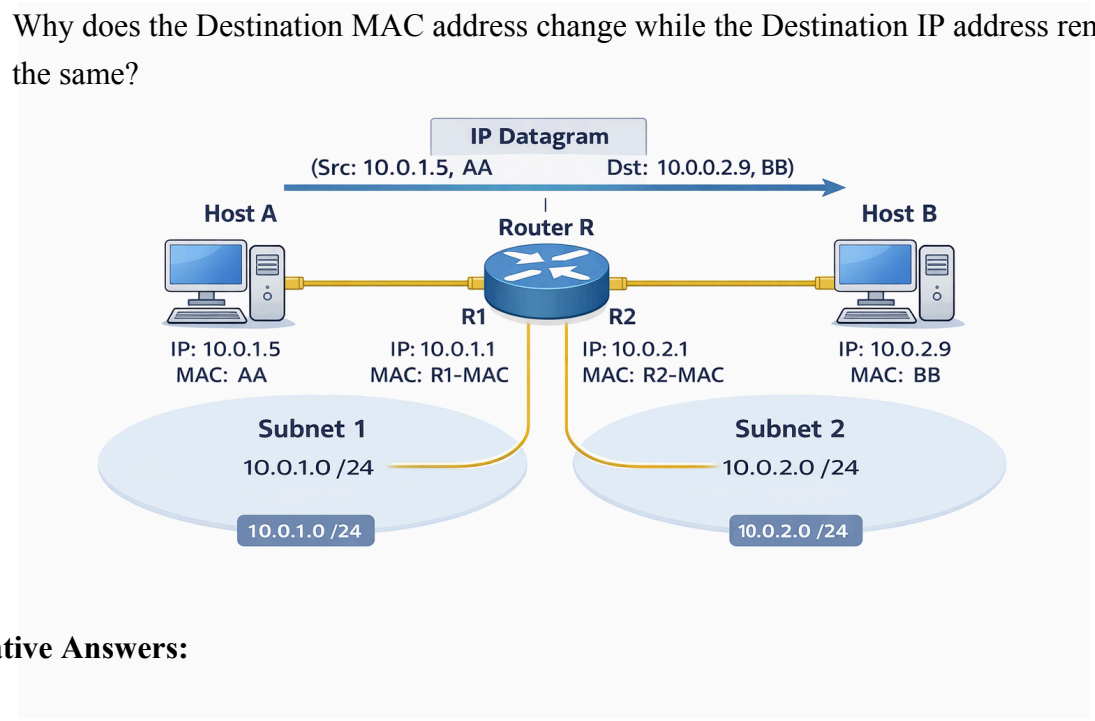
iii) (15p) Consider the following topology:

- Host A (IP: **10.0.1.5**, MAC: **AA**) is on Subnet 1.
- Router R has interface R1 (IP: **10.0.1.1**, MAC: **R1-MAC**) facing Subnet 1 and interface R2 (IP: **10.0.2.1**, MAC: **R2-MAC**) facing Subnet 2.

- Host B (IP: **10.0.2.9**, **MAC: BB**) is on Subnet 2.

Host A sends an IP datagram to Host B. Describe the addressing in the frames at two distinct points:

1. **Frame 1 (Leaving Host A)**: What are the Source IP, Dest IP, Source MAC, and Dest MAC?
2. **Frame 2 (Leaving Router R towards Host B)**: What are the Source IP, Dest IP, Source MAC, and Dest MAC?
3. Why does the Destination MAC address change while the Destination IP address remains the same?



Indicative Answers:

i. IP addresses are hierarchical and location-dependent (like a street address), allowing routing across the global Internet. MAC addresses are flat and portable (like a Social Security Number), identifying the physical hardware interface regardless of where it is.

Scenario: If you move your laptop from your home Wi-Fi to a university coffee shop Wi-Fi, your IP address changes (assigned by the new network's DHCP to match the new subnet), but your MAC address remains the same (burned into the network card).

ii.

Steps:

- A checks its ARP cache. If empty:
- A creates an ARP Request packet.

- A broadcasts this packet to the LAN.
- B receives it, recognizes its own IP, and sends an ARP Reply.
- A receives the reply and updates its ARP table.

ARP Request Frame:

- Source MAC: AA:AA
- Dest MAC: FF:FF:FF:FF:FF:FF (Broadcast).

ARP Reply Frame:

- Source MAC: BB:BB
- Dest MAC: AA:AA (Unicast).

iii.

1. **Frame 1 (Leaving Host A):**
    - **Source IP:** 10.0.1.5 (Host A)
    - **Dest IP:** 10.0.2.9 (Host B)
    - **Source MAC:** AA (Host A)
    - **Dest MAC:** R1-MAC (Router's ingress interface). *Note: A sends to its Default Gateway.*
2. **Frame 2 (Leaving Router R towards B):**
    - **Source IP:** 10.0.1.5 (Host A)
    - **Dest IP:** 10.0.2.9 (Host B)
    - **Source MAC:** R2-MAC (Router's egress interface)
    - **Dest MAC:** BB (Host B)
3. **Why MAC changes but IP doesn't:**
    - **IP** identifies the *ultimate* source and destination; these do not change during transit.
    - **MAC** addresses are for *link-layer* delivery. Frame 1 only goes from A to the Router. The Router strips the Link headers, determines the next hop, and creates a *new* Frame (Frame 2) to get from the Router to B. Thus, the link-layer addresses must update for each "hop".

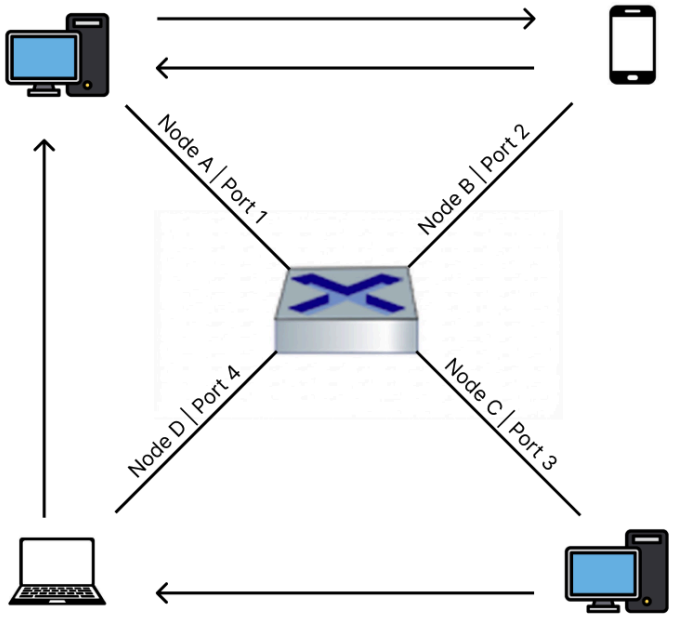
Exercise 4 (25pts)

i) (10p) Consider a switch with 4 ports (1, 2, 3, 4). The switch table is initially empty.

- Node A is on Port 1.
- Node B is on Port 2.
- Node C is on Port 3.
- Node D is on Port 4.

Trace the switch table state and the switch's forwarding actions for the following sequence of frames:

1. A sends a frame to B.
2. B replies with a frame to A.
3. C sends a frame to D.
4. D sends a frame to A.



ii) **(5p)** Compare Link-Layer Switches and Network-Layer Routers in terms of:

1. Protocol layer processed.
2. Handling of broadcast traffic (do they forward broadcasts by default?).
3. Topology structure (loops allowed vs. prevented).

iii) **(5p)** Compare **DNS**, **ARP**, and **DHCP** in terms of:

1. **Mapping Function:** Briefly describe what identifier each protocol resolves from and what it resolves to.
2. **Operational Scope:** Which of these protocols can operate globally across the Internet, and which are strictly limited to the local network? Explain the technical reason why the local protocols cannot traverse routers.

(iv) **(5p)** Ethernet switches are capable of building their forwarding tables automatically, without network administrator intervention.

1. When a frame arrives at a switch interface, what specific piece of information does the switch extract to populate its table? What mapping does it create?
2. If the switch finds no entry in its table for the frame's Destination MAC address, what action does it take?
3. Why do these table entries typically have a timer (TTL) after which they are deleted?

Indicative Answers:

i.

Step	Action	Switch Table State (After Action)	Explanation
1. A → B	Flood.	A: Port 1	Switch learns A is on Port 1. It does not know B, so it floods to 2, 3, 4.
2. B → A	Forward to 1.	A: Port 1, B: Port 2	Switch learns B is on Port 2. It knows A is on Port 1, so it forwards selectively.

3. C → D	Flood.	A: 1, B: 2, C: 3	Switch learns C is on Port 3. It does not know D, so it floods.
4. D → A	Forward to 1.	A: 1, B: 2, C: 3, D: 4	Switch learns D is on Port 4. It knows A is on Port 1, so it forwards.

ii.

Protocol Layer: Switches operate at Layer 2 (Link Layer). Routers operate at Layer 3 (Network Layer).

Broadcast Handling: Switches forward broadcast frames (like ARP requests) by default. Routers block broadcast frames (limiting the broadcast domain).

Topology: Switches generally require a loop-free topology (managed by Spanning Tree Protocol) because Layer 2 frames have no TTL (Time To Live). Routers can handle loops and mesh topologies because IP packets have TTL fields.

iii.

Mapping Function:

- **DNS:** Hostname (e.g., google.com) → IP Address (142.250.x.x).
- **ARP:** IP Address (192.168.1.5) → MAC Address (AA:BB:CC...).
- **DHCP:** None (bootstrapping) → Assigns IP, Subnet Mask, Gateway, DNS to a client.

Scope & Router Traversal:

- **Global:** DNS (Hierarchical distributed database).
- **Local:** ARP and DHCP (Discovery phase).
- **Reason:** ARP and DHCP rely on **Layer 2 Broadcasts** (FF:FF...) to function. As noted in part (ii), routers do not forward broadcast frames, so these protocols cannot naturally cross subnet boundaries without specific helpers (like DHCP Relay agents)[34].

iv.

1. **Extract:** The switch extracts the **Source MAC Address** and the **Ingress Port** number.

- **Mapping:** MAC Address → Port Number.

2. **Unknown Dest:** It **floods** the frame out all ports except the one it arrived on.
3. **TTL:** Entries have a timer because devices may move (e.g., unplugging a laptop from Port 1 and moving to Port 2) or network interface cards may be replaced. The table must stay up-to-date.

Exercise 5 (20pts)

i) (8p) A common misconception is that when you communicate with a web server (like google.com), your computer needs to know Google.com''s MAC address. In this task, you will prove that Link Layer addressing is strictly local.

1. Open your terminal/command prompt.
2. Find your Default Gateway IP address:
   - Windows: **ipconfig** (Look for "Default Gateway").
   - Mac/Linux: **netstat -nr | grep default** or **ip route**.
3. Ping your Gateway (e.g., ping 192.168.1.1) to ensure the Link Layer connection is active.
4. Ping a Remote Server (e.g., ping 8.8.8.8 or ping www.mit.edu).
5. Immediately run **arp -a** to view your Link Layer Neighbor Table.

Questions:

1. Look at your ARP table. You should see an entry (MAC address) for your Default Gateway's IP. Provide a screenshot and write it down.
2. Look for an entry for the Remote Server's IP (e.g., **8.8.8.8**). Is it there?
3. Explain your findings. When your computer creates the Ethernet frame destined for Google (**8.8.8.8**), which Destination MAC address does it use in the Ethernet header? The MAC of Google's server or the MAC of your Gateway? Why?

ii) (6p) The first 24 bits (3 bytes) of a MAC address are known as the **Organizationally Unique Identifier (OUI)** and identify the manufacturer of the network card.
Run ipconfig /all (Windows) or ifconfig / ip link (Linux/macOS) on your machine.

1. Identify the Physical Address (MAC) of your active network adapter (Wi-Fi or Ethernet). Screenshot the result, and write it down.
2. Use an online **OUI** Lookup tool (e.g., wireshark.org/tools/oui-lookup) to search for the first 3 bytes of your MAC address. Who is the manufacturer of your network interface card (e.g., Intel, Realtek, Apple)?
3. Why is it important that OUIs are globally unique?

iii) (6p) Consider the following output from a Linux netstat -i command on a busy server:

Iface	MTU	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR
eth0	1500	24050	0	0	0	23000	0
eth1	1500	85000	450	200	50	84000	5

1. Which interface (**eth0** or **eth1**) is experiencing physical layer or link layer issues?
2. **RX-ERR** often indicates "Cyclic Redundancy Check (CRC)" failures. What does a CRC failure tell you about the physical integrity of the cable or signal?
3. **RX-DRP** (Dropped packets) often increases when the driver/buffer is overwhelmed. Is this a physical cable fault or a system performance bottleneck? Explain briefly.

For this exercise, provide screenshots wherever you think it's necessary. Make sure your answers clearly indicate which section and question they correspond to.

Indicative Answers

i.

Gateway Entry: You will see an entry like 192.168.1.1 mapped to a specific MAC (e.g., a4-b1-c2...).

Remote Server (8.8.8.8): You will **NOT** see an entry for 8.8.8.8 in your ARP table.

Explanation: Link Layer addressing is **local**. Your computer knows 8.8.8.8 is not on the local subnet (using the subnet mask). Therefore, it encapsulates the IP packet for 8.8.8.8 inside an Ethernet frame addressed to your **Default Gateway's MAC address**. The Gateway is responsible for routing it further .

ii.

1. **Guarantees Unique Hardware Addresses**: By assigning a unique OUI to every manufacturer, the manufacturer only needs to ensure the last 24 bits they assign are unique within their own factory. This combination (UniqueOUI + UniqueDeviceID) mathematically guarantees that no two network cards in the world share the same MAC address.
2. **Prevents Network Collisions**: If two devices on the same local network had the same MAC address, switches would be unable to correctly forward frames (constantly updating their forwarding tables between two ports), and ARP replies would be ambiguous, causing connection failures for both devices.

iii.

1. **Interface Issues: eth1** is experiencing issues (Non-zero errors).
2. **RX-ERR (CRC Failures):** This indicates **Physical Layer** corruption. The signal is being distorted by noise, attenuation, or a bad cable/connector, causing the bits to flip during transmission so the checksum doesn't match.
3. **RX-DRP (Dropped):** This usually indicates a **System/Performance Bottleneck**. The physical frame arrived correctly (no CRC error), but the OS kernel or network driver buffer was full (computer too slow/busy) and had to discard the packet before processing it.